CS425 Final Project Proposal

Team Members: Ian St. John
Game Name: Supermassive Omega

1. Provide an overview of your game (give a short description about the game, i.e., core mechanics, game plot, objective, etc.):

   There is an old video game from my childhood called *Shooter: Starfighter Sanvein* that I really enjoyed and am going to base a lot of the mechanics off. The main mechanic is that the player must reach an exit before time runs out; however, if the player gets hit, the time goes down, and only goes back up under certain special conditions. So, there is not health, the only indicator of success is time. Also, you went through the game room by room, and ahead of time you knew the difficulty of the room and had several paths to take (the game space was a hexagonal grid, so you could approach it in many different ways).

   I really like this arcade style and think it will be fun to implement, having challenges but would not be impossible for myself to complete. My take on this game would be themed after cyber-security (think Tron) and the instead of rooms it would be computers in a network. The player would assume the role of a white-hat hacker either doing a pen test on a network or protecting the network from attack. I think this idea would mesh well with the idea of a time limit being the player health due to the time sensitive nature of network security.

2. What will be the main **technical components** (game AI, motion planning, physics, procedural content generation) that your game will focus on?

   The network that the player must move through within a given amount of time *could* be randomly generated using a (**procedural content generation**). When the player picks a node to enter, they will be faced with different malware/viruses that will behave differently (**game AI via behavior trees**). Depending on the difficulty of implementing OpenGL with SDL, the game might be 2D or 3D which will change the complexity of the **physics**. Regardless, a goal for the project that I have is to have one enemy type with boid behavior.

3. What **game engine**/tools/libraries will your game use?

   I have used several game engines in the past and this time I want to try to use the one we have been working on in class. I really want the game to be in 3D, so the difficulty will be implementing **OpenGL** with **SDL**. I have taken several OpenGL classes in the past so I have experience, but if I do not make enough progress on the implementation of 3D, I will change the scope of the game to 2D.

I have plenty of knowledge with 3D modeling software such as **Blender**, so upon implementation of OpenGL, I will use Blender to make the game models and animations. I will also be using **Visual Studio Code** to program the game as well. I prefer using Visual Studio Code over the normal full Visual Studio, because I found having to configure the compilation myself and hooking the system up to my own debugger really makes me more aware over what is happening to my data during execution.

That is also I reason I want to implement OpenGL in SDL myself instead of using an engine like Unity3D, when I have worked with those engines in the past, I found that I was restricted to work only in the workflows that was intended for that engine. I would like to try implementing my own workflows and techniques into an engine.

4. Why do you think your game is interesting/exciting?

Other than the blatant nostalgia, programming my own 3D engine is exciting for me as a developer. For players, I think in the American market outside of arcades, this time trial style shooters are rare and would be a unique experience for a lot of people.

A stretch goal for my project would be to have a slight educational aspect to it, for instance having an enemy that is called a rootkit that hides other malware. For people who do cyber security that might seem very surface level and not interesting, but if lay-people play the game, they might learn through playing the game that a rootkit hides other programs. I think that is an interesting prospect.

5. How would you divide the tasks among your team members (if more than one)?

Sadly, due to Covid finding a task member was difficult and I am on my own. But to keep myself on task, I plan to set up a Kanban to keep track of feature progress and to track development. This is going to be important for me, considering that I want to implement 3D rendering all on my own, I have to set a self-imposed deadline, and if I do not reach it, I must abandon that feature and move ahead to complete the project. That I feel is going to be the hardest part.